General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Memorandum 33-799

Integration-Free Interval Doubling for Riccati Equation Solutions

Gerald J. Bierman
Jet Propulsion Laboratory

Gursharan S. Sidhu
State University of New York at Buffalo

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

October 1, 1976

PREFACE

The work described in this report was performed by the Mission Analysis Division of the Jet Propulsion Laboratory. G. S. Sidhu's work was supported in part by an Institutional Funds Grant from the State University of New York at Buffalo.

CONTENTS

ABSTRACT

Starting with certain identities obtained by Reid and Redheffer for general matrix Riccati equations we give various algorithms for the case of constant coefficients. The algorithms are based on two ideas -- first, relate the RE solution with general initial conditions to anchored RE solutions; and second, when the coefficients are constant the anchored solutions have a basic shift-invariance property. These ideas are used to construct an integration free superlinearly convergent iterative solution to the algebraic RE. Our algorithm, arranged in square-root form, is thought to be numerically stable and competitive with other methods of solving the algebraic RE.

I. INTRODUCTION

We shall be concerned here with the solution of the following matrix Riccati differential equation (RE) which lies at the heart of many systems problems, e.g. optimal control (e.g.[3]) and least-squares estimation(e.g.[4]):

$$\frac{d}{dt} P(t) = FP(t) + P(t)F^T + Q - P(t)DP(t), \quad P(0) = \Pi_0 \tag{1}$$

In this paper F, Q, and D are assumed to be known <u>constant</u> matrices.

It has been shown in a wide body of literature that if $P_s(.)$ is the solution of

$$\frac{d}{dt} P_s(t) = FP_s(t) + P_s(t)F^T + Q - P_s(t)DP_s(t), \quad t \geq s, \quad P_s(s) = 0 \tag{2}$$

i.e. a solution of (1) anchored to zero at time s, then one can recover $P(.)$ from $P_s(.)$ through the relation

$$P(t) = P_s(t) + \Phi(t,s|P_s(.))[I + P(s)\mu(t,s|P_s(.))]^{-1}P(s)\Phi^T(t,s|P_s(.)) \tag{3}$$

where $\Phi(.,.|P_s(.))$ and $\mu(.,.|P_s(.))$ are defined by

$$\frac{\partial}{\partial t} \Phi(t,s|P_s(.)) = [F - P_s(t)D]\, \Phi(t,s|P_s(.)), \quad \Phi(s,s|P_s(.)) = I \tag{4}$$

$$\mu(t,s|P_s(.)) = \int_s^t \Phi^T(\tau,s|P_s(.))D\Phi(\tau,s|P_s(.))d\tau \tag{5}$$

Equation (3) has been known at least since the work of Sandor[5] who obtained it in a slightly different form (he assumed that P(s) is nonsingular). To the best of our knowledge the representation (3), in fact for general nonsymmetric REs, is due to Reid, see e.g. [6, eqn. 2.8]. Reid's notation for the arguments of Φ and μ has been adopted since it emphasizes the role of $P_s(.)$. In an estimation context, relations such as (3)-(5) have been discovered by Lainiotis[8] via his partitioning approach and also by Womble and Potter[9]-[10]. Lainiotis' work highlighted the importance of the smoothing error variance relation

$$P(s|t) = [P^{-1}(s) + \mu(t,s|P_s(.))]^{-1} \tag{6}$$

In this paper we review certain integration-free representations for RE solutions and use these recursions to construct a superlinearly convergent solution to the algebraic RE. Our results are expressed in algorithmic form.

II. SHIFT-INVARIANCE PROPERTY OF $P_s(.)$, $\Phi(.,.|P_s(.))$ and $\mu(.,|P_s(.))$

When F, Q, and D are constant, it has been noted independently by Womble and Potter[9]-[10], Sidhu[7,section III.8], and Lainiotis[8] that $P_s(.)$, $\Phi(.,.|P_s(.))$ and $\mu(.,.|P_s(.))$ have an important <u>shift-invariance property</u>, namely that for all $t \geq s \geq 0$,

$$P_s(t) = P_0(t - s); \quad \Phi(t,s|P_s(.)) = \Phi(t-s,0|P_0(.)); \quad \mu(t,s|P_s(.)) = \mu(t-s,0|P_0(.)) \tag{7}$$

which prompt the simpler notation $\Phi_0(t) = \Phi(t,0|P_0(.))$ and $\mu_0(t) = \mu(t,0|P_0(.))$. Thus (3) can now be written as

$$P(t) = P_0(t-s) + \Phi_0(t-s)\,[I + P(s)\mu_0(t-s)^{-1}P(s)\Phi_0^T(t-s) \qquad (8)$$

an expression that leads to the following recursive algorithm given independently by Womble and Potter[9]-[10], Sidhu [7], and Lainiotis[8]:

Algorithm 1: Recursion for $P(k\delta)$, $k = 1,2,3, \ldots$

(i) For a preselected $\delta > 0$ integrate (2), (4)-(5) to obtain $P_0(\delta)$, $\Phi_0(\delta)$, $\mu_0(\delta)$.

(ii) Compute $P(k\delta)$, $k = 1,2,3, \ldots$ recursively, starting with $P(0) = \Pi_0$, through

$$P((k+1)\delta) = P_0(\delta) + \Phi_0(\delta)[I + P(k\delta)\mu_0(\delta)]^{-1}P(k\delta)\Phi_0^T(\delta) \qquad (9)$$

Note that step (ii) is an integration-free step. Equation (9) is of course just (8) with $t = (k+1)\delta$ and $s = k\delta$. The simplicity of (9) makes it an attractive candidate for use in numerical computation. However, this algorithm has certain undesirable features which include numerical sensitivity to the initial matrix Π_0, and a lack of symmetry. One can avoid the influence of Π_0 on the recursions by computing $P_0(k\delta)$ and then obtaining $P(.)$ through the relation

$$P(k\delta) = P_0(k\delta) + \Phi_0(k\delta)[I + \Pi_0\mu_0(k\delta)]^{-1}\Pi_0\Phi_0(k\delta) \qquad (10a)$$

[this is just (8) with $s = 0$ and $t = k\delta$] or when Π_0 is nonsingular one could use the symmetric relation

$$P(k\delta) = P_0(k\delta) + \Phi_0(k\delta)[\Pi_0^{-1} + \mu_0(k\delta)]^{-1}\Phi_0^T(k\delta) \qquad (10b)$$

The result (10) is better numerically because it avoids the cumulative effects of an ill-conditioned Π_0 which can propagate through (9).

Note that (10) requires the computation of $\Phi_0(k\delta)$ and $\mu_0(k\delta)$; but these terms are of interest. Monitoring Φ_0 gives an on-line measure of stability of the time varying linear system (4); and μ_0 is useful for analyzing the advantages to be gained from smoothing.

III. RECURSIONS FOR $P_0(.)$, $\Phi_0(.)$ AND $\mu_0(.)$

It might seem that the calculation of P_0, Φ_0, and μ_0, needed in (10), would entail considerably more computation than does the recursion (9). This is in fact not the case as is shown in the following lemma due to Reid [6,p.21] and Redheffer[11], whose results have here been modified to reflect the shift-invariances (7).

Lemma 1: For $t \geq s \geq 0$ we have

$$P_0(t) = P_0(t - s) + \Phi_0(t - s)[I + P_0(s)\mu_0(t - s)]^{-1}P_0(s)\Phi_0^T(t - s), \tag{11}$$

$$\Phi_0(t) = \Phi_0(t - s)[I + P_0(s)\mu_0(t - s)]^{-1}\Phi_0(s), \tag{12}$$

$$\mu_0(t) = \mu_0(s) + \Phi_0^T(s)\mu_0(t - s)[I + P_0(s)\mu_0(t - s)]^{-1}\Phi_0(s). \tag{13}$$

More recently, this lemma has been studied in independent work by Sidhu and Desai[1] and Ljung et al.[2] where it has been shown that $P_0(.)$, $\Phi_0(.)$ and $\mu_0(.)$ can be interpreted as elements of a scattering matrix that can be used to solve two-point boundary value problems. Then the results of

the lemma have an interpretation in terms of adding layers of a scattering medium. The time-varying and nonsymmetric cases are also considered in [1]-[2].

By setting $t = (k+1)\delta$ and $s = \delta$ in these equations one can obtain recursions that provide $P_0(k\delta)$, $\Phi_0(k\delta)$ and $\mu_0(k\delta)$ at $k = 1,2,3,\ldots$; this is left to the reader (see also [16]). Instead we present a recursion that is of special value when solving the algebraic RE.

IV. AN INTERVAL DOUBLING ALGORITHM

An algorithm that recursively generates $P_0(.)$, $\Phi_0(.)$, and $\mu_0(.)$, at the points $t = 2^k\delta$, $k = 0,1,2, \ldots$, i.e. progresses geometrically (doubling the interval at each recursion), follows readily from lemma 1. For this, set $s = 2^k\delta$ and $t = 2s = 2^{k+1}\delta$ in the equations of lemma 1. Thus we have:

Algorithm 2: Interval-Doubling, Integration-Free Recursions

(i) As for algorithm 1, for a preselected $\delta > 0$, compute $P_0(\delta)$, $\Phi_0(\delta)$, $\mu_0(\delta)$ from (2), (4)-(5).

(ii) Then recursively compute $P_0(2^k\delta)$, $\Phi_0(2^k\delta)$, $\mu_0(2^k\delta)$ through the recursions

$$P_0(2^{k+1}\delta) = P_0(2^k\delta) + \Phi_0(2^k\delta)[I + P_0(2^k\delta)\mu_0(2^k\delta)]^{-1}P_0(2^k\delta)\Phi_0^T(2^k\delta) \tag{14}$$

$$\Phi_0(2^{k+1}\delta) = \Phi_0(2^k\delta)[I + P_0(2^k\delta)\mu_0(2^k\delta)]^{-1}\Phi_0(2^k\delta) \tag{15}$$

$$\mu_0(2^{k+1}\delta) = \mu_0(2^k\delta) + \Phi_0^T(2^k\delta)\mu_0(2^k\delta)[I + P_0(2^k\delta)\mu_0(2^k\delta)]^{-1}\Phi_0(2^k\delta) \tag{16}$$

When $P_0(.)$ and $\mu_0(.)$ are nonsingular matrices, the recursions can be put in a symmetric form. The symmetric form involves additional matrix inversions. However, exploiting symmetry can reduce certain of the computations and numerical errors. We omit details of such a formulation because in Section V the interval doubling algorithm is discussed using matrix square roots which implicitly preserve both symmetry and positivity of the variance matrices.

Algorithm 2 can be used to solve the algebraic RE:

$$0 = F\bar{P} + \bar{P}F^T + Q - \bar{P}D\bar{P} \tag{17}$$

since $\bar{P} = \lim_{t \to \infty} P(t)$. When $(F,Q^{1/2})$ is stabilizable and $(F,D^{1/2})$ is detectable, the limit is independent of the initial value Π_0. Using $\Pi_0 = 0$ it follows that

$$\bar{P} = P_0(\infty) = \lim_{k \to \infty} P_0(2^k\delta) \tag{18}$$

The idea of using interval doubling algorithms for solving the steady-state problem and hence the algebraic RE was proposed in [12]. There, an iterative method, quite different from the one given here, was developed for discrete-time Lyapunov and Riccati equations. Interval doubling is discussed in [1]-[2] and, as noted there, doubling methods have been used in radiative transfer problems as well.

It is interesting to note that the above results hold <u>without change</u> for the discrete-time Riccati equation, i.e. for i = 0,1,2, ...,

$$P(i+1) = \psi P(i)\psi^T + Q - [\psi P(i)H^T+C][I+HP(i)H^T]^{-1}[C^T+HP(i)\psi^T], \; P(0) = \Pi_0 \tag{19}$$

In this case, we have $\delta = 1$ and the algorithm is initialized with

$$P_0(1) = Q - CC^T; \quad \Phi_0(1) = \Psi - CH \; ; \quad \mu_0(1) = H^TH \tag{20}$$

V. A SQUARE-ROOT INTERVAL DOUBLING ALGORITHM

In most applications the RE (2) is such that $\delta > 0$ can be so chosen that $P_0(\delta)$ is a positive definite matrix. It then follows that $P_0(k\delta)$ is also positive definite for all $k > 1$; and thus we can arrange the doubling algorithm in a symmetric form which is readily put into a square root recursion. Thus, using orthogonal transformation techniques (see [13]), which have been successfully applied to least squares estimation [14]-[15], we now develop recursions for Λ_k, Γ_k, and Φ_k, where

$$P_0\ (2^k\delta) = \Lambda_k\Lambda_k^T \; ; \; \mu_0(2^k\delta) = \Gamma_k^T\Gamma_k \; ; \; \Phi_0(2^k\delta) = \Phi_k \tag{21}$$

Here, Λ_k and Γ_k are by construction upper triangular matrices. The definition of Λ_k and Γ_k is of course arbitrary, our asymmetric choice in (21) results from the observation that while $P_0(.)$ is a variance matrix, $\mu_0(.)$ is an information matrix. Further, this choice reduces the number of arithmetic operations in our square-root algorithm.

To make our square-root algorithm transparent, we start by rewriting (14)-(16)

$$\Lambda_{k+1}\Lambda_{k+1}^T = \Lambda_k\Lambda_k^T + (\Phi_k\Lambda_k)[I + X_k^TX_k]^{-1}(\Phi_k\Lambda_k)^T \tag{22}$$

$$\Gamma_{k+1}^T\Gamma_{k+1} = \Gamma_k^T\Gamma_k + (\Gamma_k\Phi_k)^T[I + X_kX_k^T]^{-1}(\Gamma_k\Phi_k) \tag{23}$$

$$\Phi_{k+1} = \Phi_k \Gamma_k^{-1} [I + X_k X_k^T]^{-1} \Gamma_k \Phi_k \tag{24}$$

where

$$X_k = \Gamma_k \Lambda_k$$

The matrix inversions appearing in (22)-(24) can be avoided by utilizing certain properties of orthogonal transformations:

Lemma 2: Let S be an orthogonal matrix chosen so that

$$n \left\{ \left[\overbrace{I_n}^{n} \mid \overbrace{X}^{p} \right] S = \left[\overbrace{R}^{n} \mid \overbrace{0}^{p} \right] \right. \tag{25}$$

where I_n is the $n \times n$ identity matrix, and S is partitioned consistent with (25) as

$$S = \left(\begin{array}{c|c} S_{11} & S_{12} \\ \hline S_{21} & S_{22} \end{array} \right)$$

Then,

$$[I_n + XX^T]^{-1} = S_{11} S_{11}^T \; ; \quad [I_p + X^T X]^{-1} = S_{22} S_{22}^T \tag{26}$$

Proof: From (25) and the orthogonality of S it follows that $I_n = RS_{11}^T$ and $X = RS_{21}^T$. Thus, $R = S_{11}^{-T}$ and $X = S_{11}^{-T} S_{21}^T$. Also, from (25) we have $[I_n + XX^T] = RR^T$, and hence the first of (26) follows immediately. Next, we note that since S is orthogonal, $S_{22}^T S_{21} + S_{12}^T S_{11} = 0$ and $S_{22}^T S_{22} + S_{12}^T S_{12} = I_p$; hence,

$S_{22}^T(I_p + X^TX)S_{22} = S_{22}^T S_{22} + S_{22}^T S_{21} S_{11}^{-1} S_{11}^{-T} S_{21} S_{22} = I_p$, which immediately gives us the other relation in (26).

Having established this lemma it is easy to obtain the following:

Algorithm 3: Square-Root Interval-Doubling Algorithm

(i) Construct an orthogonal $S^{(k)}$ such that

$$[\ I_n \mid \Gamma_k \Lambda_k\ ]\ S^{(k)} = [R^{(k)} \mid 0\ ] \tag{27}$$

where $R^{(k)}$ is upper triangular.

(ii) By using implicitly defined orthogonal transformations $S_\Lambda^{(k)}$ and $S_\Gamma^{(k)}$ construct upper triangular matrices Λ_{k+1} and Γ_{k+1} such that

$$[\ \Lambda_k \mid \Phi_k \Lambda_k S_{22}^{(k)}\ ]\ S_\Lambda^{(k)} = [\Lambda_{k+1} \mid \underbrace{0}_{n}\ ] \tag{28}$$

$$S_\Gamma^{(k)} \begin{pmatrix} \Gamma_k \\ \hline Y_k \end{pmatrix} = \begin{pmatrix} \Gamma_{k+1} \\ \hline 0 \end{pmatrix} \Big\} n \tag{29}$$

where $Y_k = (S_{11}^{(k)})^T \Gamma_k \Phi_k$

(iii)
$$\Phi_{k+1} = \Phi_k \Gamma_k^{-1} S_{11}^{(k)} Y_k \tag{30}$$

Note that significant computational saving can be realized by exploiting the triangular nature of Λ_k, Γ_k, and $S_{11}^{(k)}$. Details of the orthogonal transformation can be found in [13] and [14].

REFERENCES

1. G. S. Sidhu and U. B. Desai, "A Method of Orthogonal Directions--Part II: TPBVPs, Scattering Matrices, and Estimation Algorithms," Tech. Report, Dept. of Electrical Engrg., State Univ. of New York at Buffalo, May 1975; also submitted for journal publication.

2. L. Ljung, T. Kailath, and B. Friedlander, "Scattering Theory and Linear Least-Squares Estimation, Part I: Continuous-Time Problems," Proc. of the IEEE, vol. 64, No. 1, pp. 131-139, 1976.

3. B. D. O. Anderson and J. B. Moore, Linear Optimal Control, Prentice-Hall, 1971.

4. A. H. Jazwinski, Stochastic Processes and Filtering Theory, Academic Press, New York, 1970.

5. S. Sandor, "Sur l'équation différentielle matricielle de type Riccati," Bull. Math. Soc. Sci. Math. Phys. R. P. Roumaine(N.S.), vol.3, pp. 229-249, 1959.

6. W. T. Reid, Riccati Differential Equations, Academic Press, New York, 1972.

7. G. S. Sidhu, "A Shift-Invariance Approach to Fast Estimation Algorithms," Ph.D. thesis, Dept. of Elec. Engrg., Stanford Univ., Stanford, Calif. 94305, March 1975.

8. D. G. Lainiotis, "Partitioned Estimation Algorithms, II: Linear Estimation," Info. Sci., vol. 7, pp. 317-340, 1974.

9. M. E. Womble, "The Linear-Quadratic-Gaussian Problem with Ill-Conditioned Riccati Matrices," Ph.D. thesis, Dept. of Aero. and Astro., Mass. Inst. of Tech., Cambridge, Mass., May 1972.

10. M. E. Womble and J. E. Potter, "A Prefiltering Version of the Kalman Filter with New Numerical Integration Formulas for Riccati Equations," Proc. 1973 IEEE Conf. on Dec. and Control, San Diego, pp. 63-67, 1973.

11. R. M. Redheffer, "On the Relation of Transmission-Line Theory to Scattering and Transfer," J. Math. and Physics, vol. 41, pp. 1-41, 1962.

12. G. J. Bierman, "Steady-State Covariance Computation for Discrete Linear Systems," Proc. 1971 Joint Auto. Control Conf., St. Louis, Miss., pp. 811-819, 1971.

13. C. L. Lawson and R. J. Hanson, Solving Least-Squares Problems, Prentice Hall, Englewood Cliffs, New Jersey, 1974.

14. G. J. Bierman, "Computational Aspects of Discrete Sequential Estimation," Jet Propulsion Laboratory Rept. 900-661, May 1974 (JPL internal document).

15. G. J. Bierman, "Sequential Square Root Filtering and Smoothing of Discrete Linear Systems," Automatica, vol. 10, pp. 147-158, 1974.

16. G. J. Bierman and G. S. Sidhu, "Some Estimation Formulae for Continuous-Time-Invariant Linear Systems," Proc. San Diego Nonlinear Estimation Symposium, Sept. 1975.

APPENDIX

COMPUTER MECHANIZATION OUTLINE FOR SQUARE-ROOT INTERVAL DOUBLING

INPUT: $\Gamma(N,N)$ $\Lambda(N,N)$ Upper Triangular Matrix Factors
$\phi(N,N)$ Transition

OUTPUT: Γ, Λ, ϕ, updated

PROGRAM MATRICES:

$$\bar{\Lambda} = \begin{bmatrix} \overbrace{\text{WORK}}^{N} \\ \hline \Lambda^T \end{bmatrix} \begin{matrix} \}N \\ \}N \end{matrix} \quad , \quad \bar{\Gamma} = \begin{bmatrix} \overbrace{\Gamma}^{N} \\ \hline \text{WORK} \end{bmatrix} \begin{matrix} \}N \\ \}N \end{matrix}$$

$$W = \begin{bmatrix} \overbrace{W_{11}}^{N} & \overbrace{W_{12}}^{N} & \overbrace{W_{13}}^{N} \\ \hline W_{21} & W_{22} & W_{23} \end{bmatrix} \begin{matrix} \}N \\ \}N \end{matrix}$$

MECHANIZATION SKETCH USING QUASI-FORTRAN

```
   DO 2  I = 1, 2N
   DO 1  J = 1, 3N
1  W(I,J)=0.
2  W(I,I)=1.
```

$W = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$

```
   DO 3  J = 2N + 1, 3N
3  W(J - N,J) = 1.
```

$W_{23} = I$

```
   DO 5  I = 1,N
   DO 5  J = I,N
   σ = 0.
   DO 4  K = I,J
4  σ = σ + Γ̄(J,K)* Λ̄(N+I,K)
5  W(I,J + 2N) = σ
```

$W_{13} = X^T = (\Gamma\Lambda)^T$

NOTE $\bar{\Lambda}(N + I,J) = \Lambda(J,I)$

CALL HHL (W, 2N, 2N, 3N, N)

$$\begin{bmatrix} I & 0 & X^T \\ 0 & I & I \end{bmatrix} \xrightarrow{\text{HHL}} \begin{bmatrix} S_{11}^T & S_{21}^T & 0 \\ S_{21}^T & S_{22}^T & R^T \end{bmatrix}$$

This is step i, page 9

```
    DO 7  J = 1,N
    DO 7  I = 1,N
    σ = 0.
    DO 6  K = 1,I
 6  σ = σ + Ā(N + I,K)Kφ(J,K)
 7  W(I,J + 2N) = σ
```

$W_{13} = \Lambda^T \phi^T$

```
    DO 9  J = 1,N
    DO 9  I = 1,N
    σ = 0.
    DO 8  K = 1,N
 8  σ = σ + W(N + I, N + K)*W(K, 2N + J)
 9  Ā(I,J) = σ
```

$\bar{\Lambda}_1 = S_{22}^T \Lambda^T \phi^T$

CALL HHL (Λ, 2N, N, N, N)

$$\begin{bmatrix} S_{22}^T & \Lambda^T \phi^T \\ & \Lambda^T \end{bmatrix} \xrightarrow{\text{HHL}} \begin{bmatrix} 0 \\ \Lambda^T \end{bmatrix} \qquad \text{Eq. (28)}$$

```
    DO 11  I = 1,N
    DO 11  J = I,N
    σ = 0.
    DO 10  K = I,J
10  σ = σ + W(I,K)* Γ̄(K,J)
11  Λ(I,J) = σ
```

$\Lambda_1 = S_{11}^T \Gamma$ (upper triangular)

```
    DO 13  J = 1,N
    DO 13  I = 1,N
    σ  = 0.
    DO 12  K = I,N
12  σ  = σ + Ā(I,K)* φ(K,J)
13  Γ̄(N + I,J) = σ
```

$\Gamma_2 = Y = S_{11}^T \Gamma \phi$

```
    DO 15  J = 1,N
    DO 15  I = 1,N
    σ  = 0.
    DO 14  K = 1,I
14  σ  = σ + W(K,I)* Γ̄(N + K,J)
15  Ā(I,J) = σ
```

$\Lambda_1 = S_{11} Y$

```
CALL UIN(Γ, 2N, W, 2N, N)
```

$W_{11} = \Gamma^{-1}$

```
    DO 17  J = 1,N
    DO 17  I = 1,N
    σ  = 0.
    DO 16  K = 1,J
16  σ  = σ + ϕ(I,K)*W(K,J)
    W(I,N+J) = σ
```

$W_{12} = \phi\Gamma^{-1}$

```
CALL HHU(Γ, 2N, N, N, N)
```

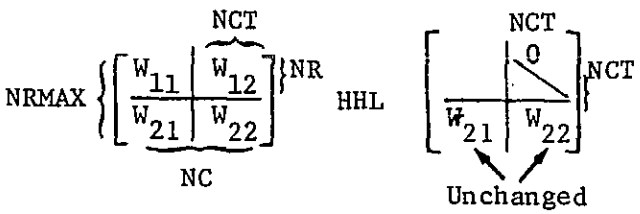
$$\begin{bmatrix} \Gamma \\ Y \end{bmatrix} \xrightarrow{\text{HHL}} \begin{bmatrix} \Gamma \\ 0 \end{bmatrix}$$

Updated Γ, Eq. (29)

```
    DO 19  J = 1,N
    DO 19  I = 1,N
    σ  = 0.
    DO 18  K = 1,N
18  σ  = σ + W(K,N+K)* Ā(K,J)
19  ϕ(I, J) = σ
```

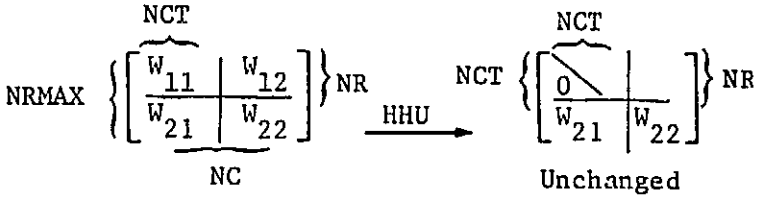
Updated ϕ, Eq. (30)

```
      SUBROUTINE HHL(W, NRMAX, NR, NC, NCT)

c  ** Lower Triangularization of W * * (SEE FIGURE)
c  W(NR, NC), with NRMAX row dimension (NR .LE. NRMAX)
c  NCT .LE. NR,  NCT .LE.  NC
c  TW computed, with T orthogonal.  The result, a partially lower
c  triangular matrix is stored back in the bottom part of W
c  Dimension of low triangular matrix is NCT

      DIMENSION  W(NRMAX, NC)
      DOUBLE PRECISION DELTA
      DATA Z/0./, ONE/1./
      NRPNC = NR + NC              @ 'NR' PLUS 'NC'
      NST = NC - NCT + 2

      DO  50  J = NC, NST, -1
      Delta = Z
      JDIAG = NRPNC - J
      DO  10  I = 1, JDIAG
10    DELTA = DELTA + W(I,J)**2
      SIG = SQRT ( DELTA )
      IF (W(JDIAG, J) .GT.Z)  SIG = SIG

      W(JDIAG,J) = W(JDIAG,J) - SIG
      ALFA = ONE/SIG*W(JDIAG,J))
      IF (J.EQ.1) GO TO 35
      JM1 = J - 1
      DO  30  K = 1, JM1
          DELTA = Z
          DO  20  I = 1, JDIAG
20        DELTA = DELTA + W(I,K)*W(I,J)
          DELTA = DELTA*ALFA
      DO  30  I = 1, JDIAG
30    W(I,K) = W(I,K) + DELTA*W(I,J)

35    DO  40  I = 1, JDIAG
40    W(I,J) = Z
      W(I,JDIAG) = SIG
50    CONTINUE
```



FIGURE

```
      SUBROUTINE HHU(W, NRMAX, NR, NC, NCT)

c     * * Upper Triangularization of W * *  (SEE FIGURE)
c     W(NR,NC), with NRMAX row dimension (NR .LE. NRMAX)
c     The first NCT columns are triangularized.
c     T W computed, with T orthogonal.  The result, a partially
c     upper triangular matrix is stored back in the top part of W.

      DIMENSION W(NRMAX, NC)
      DOUBLE PRECISION DELTA
      DATA Z/0./, ONE/1./

      DO  40  J = 1, NCT
          DELTA = Z
          DO  10  I = J, NR
10        DELTA = DELTA + S(I,J)**2
          SIG = SQRT ( DELTA )
          IF W(J,J) .GT. Z) SIG = -SIG
      W(J,J) = W(J,J) - SIG
      ALFA = ONE/(SIG*W(J,J))
      JP1 = J+1

      DO  30  K = JP1, NC
          DELTA = Z
          DO  20  I = J, NR
20        DELTA = DELTA + W(I,K)*W(I,J)
          DELTA = DELTA*SIG

      DO  30  I = J, NR
30    W(I,K) = W(I,K) + DELTA*W(I,J)

      DO  35  I = JP1, NC
35    W(I,J) = Z
      W(J,J) = SIG
40    CONTINUE
```



FIGURE

```
      SUBROUTINE UIN(W, NWMAX, WINV, NWINM, N)

      DIMENSION W(NMAX,N), WINV(NWINM, N)

c     W upper triangular
c     WINV = W INVERSE COMPUTED
c     WINV can replace W

      DOUBLE PRECISION σ
c     It is good practice to do matrix inversion in double precision

      WINV(1,1) = 1./W(1,1)
      DO  20  J = 2,N
      WINV(J,J) = 1./W(J,J)
      JM = J-1
      DO  20  K = 1, JM
      σ = 0.
      DO  10  I = K, JM
10    σ = σ- WINV(K,I*W(I,J)
20    WINV(K,J) = σ*WINV(J,J)
      RETURN
```


NASA—JPL—Coml, L.A., Calif